

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH
TEHNOLOGIJA OSIJEK

Sveučilišni studij

MIKROUPRAVLJAČKI SUSTAV KUTIJE ZA SNIMANJE
FOTOGRAFIJA

Diplomski rad

Ivan Reponj

Osijek, 2016.



Obrazac D1: Obrazac za imenovanje Povjerenstva za obranu diplomskog rada

Osijek, 12.07.2016.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za obranu diplomskog rada

Ime i prezime studenta:	Ivan Reponj
Studij, smjer:	Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo
Mat. br. studenta, godina upisa:	D-619R, 23.10.2013.
Mentor:	Prof.dr.sc. Željko Hocenski
Sumentor:	Doc.dr.sc. Ivan Aleksi
Predsjednik Povjerenstva:	Doc.dr.sc. Ivan Aleksi
Član Povjerenstva:	Dr.sc. Ivan Vidović
Naslov diplomskog rada:	Mikroupravljački sustav kutije za snimanje fotografija
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak diplomskog rada:	U ovom radu potrebno je izraditi kutiju za snimanje fotografija s upravljam elementima te difuznom rasvjetom. Potrebno je realizirati računalom upravljano podnožje s podesivom orijentacijom predmeta. Potrebno je omogućiti udaljeno upravljanje rasvjetom te usporediti utjecaje različitih difuznih materijala na osvjetljenje objekata.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 Postignuti rezultati u odnosu na složenost zadatka: 2 Jasnoća pismenog izražavanja: 2 Razina samostalnosti: 2
Datum prijedloga ocjene mentora:	12.07.2016.
Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:



ETFOS
ELEKTROTEHNIČKI FAKULTET OSIJEK



Sveučilište Josipa Jurja Strossmayera u Osijeku

IZJAVA O ORIGINALNOSTI RADA

Osijek, 15.07.2016.

Ime i prezime studenta:

Ivan Reponj

Studij:

Diplomski sveučilišni studij Računarstvo, smjer Procesno računarstvo

Mat. br. studenta, godina upisa:

D-619R, 23.10.2013.

Ephorus podudaranje [%]:

4

Ovom izjavom izjavljujem da je rad pod nazivom: **Mikroupravljački sustav kutije za snimanje fotografija**

izrađen pod vodstvom mentora Prof.dr.sc. Željko Hocenski

i sumentora Doc.dr.sc. Ivan Aleksi

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija. Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET OSIJEK

IZJAVA

Ja, Ivan Reponj, OIB: 11454022492, student/ica na studiju: Diplomski sveučilišni studij

Računarstvo, smjer Procesno računarstvo, dajem suglasnost Elektrotehničkom fakultetu

Osijek da pohrani i javno objavi moj **diplomski rad**:

Mikroupravljački sustav kutije za snimanje fotografija

u javno dostupnom fakultetskom, sveučilišnom i nacionalnom repozitoriju.

Osijek, 15.07.2016. 2016.

potpis

Sadržaj

1. UVOD	1
2. ARDUINO MIKROUPRAVLJAČ	2
2.1. Arduino Nano	3
2.2. Pinovi.....	5
2.3. Modulacija širine impulsa (PWM)	6
3. IZRADA PROJEKTA.....	8
3.1 Praktični rad.....	8
3.2. Shema i elektronika	10
3.4. Programski kod.....	12
3.5. Moguća poboljšanja.....	15
4. FOTOGRAFIRANJE STAKLENIH BOCA I ČAŠA PIĆA	16
4.1. Gradijent refleksija sa velikom softboksom	16
4.2. Kvalitetno fotografiranje proizvoda	17
4.3. Osvjetljavanje čaše vina (slikanje posljednje kapljice)	18
4.4. Fotografiranje boca i pića	19
4.5. Fotografiranje čaše i boce piva.....	20
4.6. Efekt hladne boce	22
4.7. Pleksiglas za fotografiranje proizvoda	23
4.8. Deset bitnih stvari za dobru fotografiju vina	24
4.9. Korištenje kutije za fotografiranje	25
5. ZAKLJUČAK	28
LITERATURA.....	29
SAŽETAK.....	30
ABSTRACT	31
ŽIVOTOPIS	32
PRILOG	33

1. UVOD

Cilj ovog diplomskog rada je izraditi mikroupravljačku kutiju za fotografiranje boca i čaša pića. Koristit će se Arduino platforma otvorenog koda sa sklopovljem i programskim paketom. U ovom slučaju koristit će se za kontrolu rotacije podnožja motorom na kojem se fotografiraju pića i za rasvjetu koja obuhvaća bočna LED svjetla čiji intenzitet treba biti upravljiv po želji te prednje usmjereno halogeno svjetlo. Kako bi sustav bio upravljiv potrebno je spojiti elektroniku i isprogramirati mikroupravljač. Slikanje fotografija namijenjeno je promotivnom sadržaju različitih boca pića i čaša bez dodatne obrade. Praktični dio rada obuhvaća izradu kutije za snimanje fotografija i njenu rasvjetu u kombinaciji sa spomenutim Arduino upravljanjem. U pismenom radu napraviti će se osvrt na Arduino mikroupravljače i korišteni Arduino Nano. Detaljno će biti opisan postupak izrade praktičnog dijela rada, te elektronska shema i programski kod sustava. Opisat će se razne tehnike fotografiranja pića, problemi rasvjete i refleksije svjetla i objasniti potrebni pojmovi i korištena elektronika. Oprema će se koristiti za fotografiranje i opisat će se pokusne fotografije.

2. ARDUINO MIKROUPRAVLJAČ

Arduino je alat za svakoga tko želi na vlastiti način napraviti kreativnu interakciju sa svijetom oko sebe. Veoma se brzo uči, jednostavan je za korištenje i može ga koristiti bilo tko, od školaraca do umirovljenika. Može se dobiti velik broj rezultata i djelovati na različite elektroničke uređaje i komponente. LED svjetlima i motorima kao u ovom diplomskom radu, internetom, pametnim telefonima, kućanskim aparatima, *Bluetooth* komunikacijom, stvaranjem interneta stvari itd. Arduino zajednica velikog broja ljudi obuhvaća osobe koje se bave Arduinom, njihove radove, ideje, projekte, upute i savjete objavljene na internetu. Zato svaka ideja za izradu vlastitog projekta daje velike šanse da je to netko već napravio i stavio na internet sve vezano uz to. Sheme spajanja, programske kodove, slike i videe.

Arduino je softverska kompanija i korisnička zajednica koja dizajnira i proizvodi otvorenu platformu softvera i hardvera, mikroupravljačku opremu za stvaranje digitalnih uređaja i interaktivnih objekata koji mogu osjetiti i kontrolirati fizičke uređaje. Arduino je platforma koja konstruktorima i dizajnerima omogućuje stvaranje uređaja i naprava koji se preko računala kontroliraju u fizičkom svijetu. Služi za kreiranje elektroničkih projekata pomoću hardvera koji predstavlja fizički mikroupravljač koji je programibilni strujni krug i softvera koji se programira i pokreće sa računala kako bi se upravljala mikroupravljačka pločica. Pločicu su kreirali studenti u Italiji 2005. godine i popularna je zbog svoje jednostavnosti. Ostali mikroupravljački sustavi zahtijevaju hardver (programer) za programiranje, ali za Arduino je dovoljan samo USB kabal da bi se povezo s bilo kojim računalom sa bilo kojim operacijskim sustavom.

Arduino pločice mogu čitati ulaze (senzor za svjetlo, pritisak prekidača ili Twitter poruku) i pretvoriti ih u izlaz (pokrenuti motor, upaliti LED svjetlo, objaviti nešto na internetu). Moguće je zapovijedati svojoj pločici što da radi tako da se pošalje skup instrukcija mikroupravljaču na pločici. Za to se koristi Arduino programibilni jezik (baziran na *Wiring* otvorenoj programskoj platformi) i Arduino softver (IDE, engl. *Integrated Development Environment*) baziran na *Processing-u*. Arduino je dosad bio temelj za tisuće projekata sa svakodnevnim predmetima i složenim znanstvenim instrumenatima. Arduino zajednica ljudi diljem svijeta okupljena je oko ove otvorene platforme i omogućuje pristup velikim količinama znanja dostupnog početnicima ali i stručnjacima.

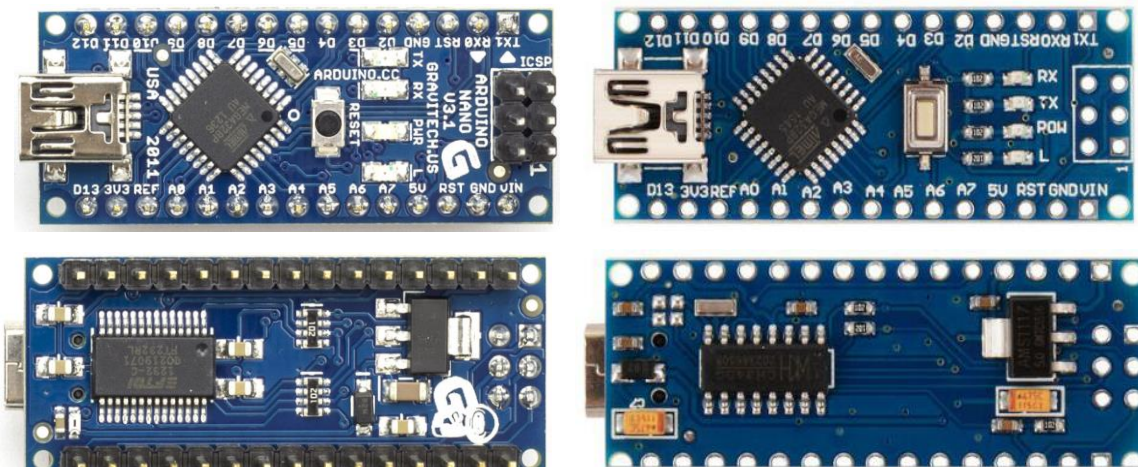
Arduino je nastao na Ivrea institutu za interakcijski dizajn kao jednostavan alat za brzu izradu prototipa usmjeren na studente bez znanja o elektronici i programiranju. Kako se ideja proširila, Arduino pločica prilagodila se novim potrebama i izazovima. Od jednostavnih 8-bitnih pločica do aplikacija interneta stvari.

Zašto koristiti baš Arduino? Zahvaljujući jednostavnom i pristupačnom korisničkom iskustvu Arduino je korišten u tisućama različitih projekata i aplikacija. Arduino softver je jednostavan za početnike i dovoljno prilagodljiv za napredne korisnike te radi na Mac-u, Windows-u i Linux-u. Arduino potiče kreativnost i učenje novih stvari. Bilo tko, djeca, umjetnici i programeri mogu nešto napraviti jer je dovoljno pratiti upute korak po korak ili mogu podijeliti svoju ideju sa članovima Arduino zajednice na internetu. Postoje i druge mikroupravljačke platforme dostupne za fizičko računalstvo kao što su *Parallax Basic Stamp*, *Netmedia's BX-24*, *Phidgets*, *MIT's Handyboard*, itd. Svi oni, kao i Arduino, pretvaraju komplicirane detalje mikroupravljačkog programiranja u jednostavno korištenje. Arduino pojednostavljuje rad s mikroupravljačima i nudi neke prednosti za nastavnike, studente i zainteresirane amatere u drugim područjima.

Arduino je relativno jeftin u odnosu na druge mikroupravljačke platforme. Najjeftinije verzije Arduino modula mogu se izraditi ručno, a najskuplji moduli koštaju manje od pedeset dolara. Većina mikrokontrolerskih sustava ograničeni su na Windows, dok Arduino radi na svim operacijskim sustavima. Ima jednostavno i jasno programsko okruženje. Arduino softver je jednostavan i za početnike i programski jezik otvorenog koda dostupan za proširenje. Jezik se može proširiti kroz C++ biblioteke. Ljudi koji žele razumjeti detalje mogu se prebaciti iz Arduina na AVR C programski jezik na kojem je i temeljen. Također se može dodati AVR-C kod u Arduino programe. Isto je i sa hardverom. Arduino pločice objavljene su pod *Creative Common* licencom pa iskusni dizajneri mogu napraviti svoje bolje verzije modula. Čak i neiskusni korisnici mogu napraviti svoje makete modula da bi razumjeli kako radi i uštedjeli novac.

2.1. Arduino Nano

U ovom diplomskom radu korištena je nebrendirana kopija Arduina Nano koja na *eBay*-u košta dva dolara. Pošto su original i kopija gotovo identični po dizajnu, specifikacijama i namjeni, u nastavku se spominje samo Arduino Nano. Korištena verzija je Nano V3.0 ATmega328P.



Sl. 1.1. *Arduino Nano (lijevo) i nebrendirana kopija (desno) prikazani odozgo i ispod.*

Arduino Nano je mala, upotpunjena pločica bazirana na ATmega 328 (Arduino Nano 3.x verzije) ili ATmega168 (Arduino Nano 2.x verzije) mikroupravljaču. Ima gotovo jednaku funkcionalnost kao Arduino Duemilanove, samo u drugačijem paketu. Nedostaje jedino ulaz za istosmjerno napajanje i radi s Mini-B USB kabelom umjesto sa klasičnim standardnim USB kabelom. Nano je dizajnirao i proizvodi ga *Gravitech*.

Arduino Nano ima sljedeće tehničke podatke. Mikroupravljač Atmel ATmega168 ili ATmega328 ima radni napon (razina logike) od 5V. Ulazni napon (preporučeno) je 7-12 V, a krajnji ulazni napon je 6-20 V. Ima 14 digitalnih ulazno/izlaznih pinova od čega 6 pružaju PWM izlaz. Ima 8 analognih ulaznih pinova. Istosmjerna struja po ulazno/izlasnom pinu je 40 mA. Fleš memorija za pohranu koda iznosi 16 KB (ATmega168) ili 32 KB (ATmega328) od kojih se 2 KB koriste za *bootloader*. SRAM memorije ima 1 KB za ATmega168 ili 2 KB za ATmega328, a EEPROM memorije 512 bajtova (ATmega168) ili 1 KB (ATmega328). Radni takt mu je 16 MHz. Širok je 18 mm, dugačak 45 mm, a masa mu je 5 g.

Arduino Nano se može programirati Arduino softverom u čijem „Tools“ izborniku biramo mikrokontroler koji koristimo (Tools > Board > Arduino Duemilanove or Nano w/ ATmega328). ATmega168 ili ATmega328 na Arduino Nano dolaze sa snimljenim *bootloader*-om koji omogućuje stavljanje novog koda na Nano bez korištenja vanjskog hardverskog programera. Moguće je i zaobići *bootloader* i mikroupravljač programirati preko ICSP (engl. *In-Circuit Serial Programming*) koristeći Arduino ISP (engl. *In-System-Programmer*).

Nano omogućuje automatski softverski reset umjesto fizičkog pritiskanja tipke reseta prije prijenosa koda. Omogućuje reset pomoću softvera na spojenom računalu. Softver omogućuje prijenos koda klikom *Upload* gumba. Kad je Nano spojen na računalo sa Mac operacijskim sustavom ili Linux-om, resetira se svaki put kada uspostavi konekciju sa softverom preko USB konekcije.

Također postoji i Croduino što je hrvatska verzija Arduina koju predstavlja Arduino Hrvatska na svojoj internet stranici e-radionica.com. To je stranica za jednostavno učenje i kupovinu elektronike u Hrvatskoj. Osim Croduina može se naći i mnoštvo senzora i aktuatora te tutorijala i primjera projekata. Kao što je to spomenuto Arduino je otvorena platforma pa su dozvoljene izmjene i nove verzije pločica. Croduino Basic je prva hrvatska Arduino kompatibilna pločica s mikrokontrolerom bazirana na Arduinu. Identičan je kao Arduino Nano, ali zbog čestog korištenja Arduina Uno napravljen je manji i sa dva pina više. Croduino je potpuno Arduino kompatibilan i sve na čemu radi Arduino radit će i sa Croduino.

2.2. Pinosi

Svaki od 14 digitalnih pinova na Arduinu Nano može se koristiti kao ulaz ili izlaz koristeći *pinMode()*, *digitalWrite()* i *digitalRead()* funkcije. Oni rade na 5 volta. Svaki pin može dati ili primiti maksimalno 40 mA struje i ima interni *pull-up* otpornik (tvornički isključen) od 20 do 50 kΩ.

Serijski: 0 (RX) i 1 (TX) koriste se za primanje (RX) i prijenos (TX) TTL serijskih podataka.

Vanjski prekidi: 2 i 3 su pinovi koji se mogu konfigurirati za pokretanje prekida na niskim vrijednostima, na rastući ili padajući brid ili promjene vrijednosti.

PWM izlazi 3, 5, 6, 9, 10 i 11 daju 8-bitni PWM izlaz sa *analogWrite()* funkcijom.

SPI pinovi 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) podržavaju SPI komunikaciju, koja, iako postoji na hardveru, trenutačno nije uključena u Arduino jeziku.

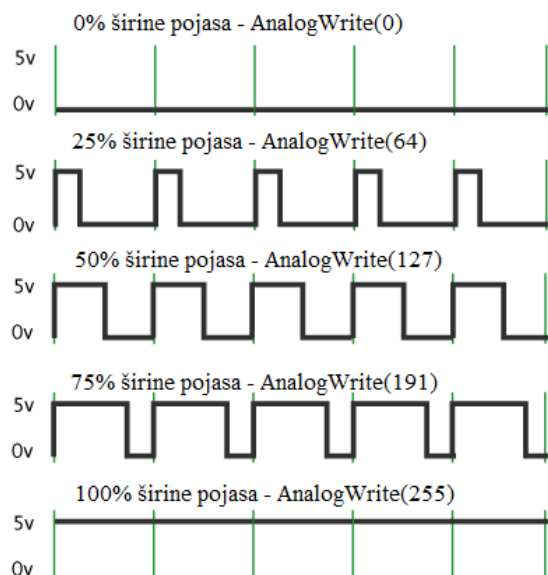
LED 13 je uzgrađeno LED svjetlo spojeno na digitalni pin 13. Kada je pin na visokoj vrijednosti (*HIGH*), LED svijetli, a kada je na niskoj vrijednosti tada je ugašeno.

Nano ima 8 analognih ulaza od kojih svaki osigurava 10 bita rezolucije, što je 1024 različitih vrijednosti. Tvornički zadano oni su od uzemljenja do 5 volta, ali je moguće pomaći gornju granicu koristeći *analogReference()* funkciju. Analogni pinovi 6 i 7 ne mogu se koristiti kao digitalni. A4 (SDA) i A5 (SCL) pinovi su podrška I²C (TWI) komunikaciji pomoću Wire biblioteke. AREF je referentni napon za analogne ulaze. Koristi se uz *analogReference()* funkciju. Tipka *Reset* pritišće se kako bi se resetirao mikroupravljač. Obično se koristi za dodavanje reset tipke.

Arduino Nano može komunicirati sa računalom, drugim Arduinoom ili drugim mikroupravljačima. ATmega168 i ATmega328 pružaju UART TTL (5V) serijsku komunikaciju dostupnu na digitalnim pinovima 0 (RX) i 1 (TX). FTDI (engl. *Future Technology Devices International*) FT232RL na pločici uspostavlja serijsku komunikaciju preko USB i FTDI drajvera (koji je u Arduino softveru). Arduino softver omogućuje slanje i primanje jednostavnih tekstualnih naredbi. RX i TX LED svjetla na pločici bljesnut će kada se podaci prenose preko FTDI i USB priključka na računalo.

2.3. Modulacija širine impulsa (PWM)

Modulacija širine impulsa ili PWM (engl. *Pulse Width Modulation*) je tehnika upravljanja analognih krugova njihovim digitalnim izlazima. Ovdje se koristi za kontrolu intenziteta LED rasvjete, ali se može koristiti i za druge stvari kao što je promjena brzine vrtnje motora. Digitalna kontrola koristi se za dobivanje pravokutnog signala koji mjenja vrijednosti iz nule u neku gornju vrijednost što kod Arduina simulira promjenu napona od 0 i 5V mjenjanjem trajanja koje signal provede na tim vrijednostima naizmjenično. Trajanje signala na gornjoj vrijednosti zove se širina pojasa koja se mjenja kako bi se dobile različite analogne vrijednosti. Ako se ta naizmjenična promjena vrijednosti signala ponavlja dovoljno brzo rezultat je stabilan napon između 0 i 5V za kontrolu intenziteta LED svjetla. Ako želimo da LED svjetlo svijetli na pola jačine smanji se struja kroz LED stvarajući kvadratni signal sa 50% radnog ciklusa. Trajanje perioda proporcionalno je PWM frekvenciji. PWM se kod Arduina provodi pomoću funkcije *analogWrite(pin, dutyCycle)*. Ona ima vrijednosti 0 do 255 tako da *analogWrite(255)* zahtjeva širinu pojasa od 100% (signal je uvijek 5V), a *analogWrite(127)* je 50% širine pojasa.



Sl. 2.1. Širine pojasa kod modulacije širine impulsa

U ovom radu prema kodu u prilogu A.2. *analogWrite()* funkcija stvara PWM izlaze D3 i D5 koji signalom aktiviraju tranzistore preko kojih se pale LED trake. Ona analognu vrijednost (PWM pravokutni signal) upisuje u pin. Nakon poziva funkcije, pin generira stabilan kvadratni signal određenog radnog ciklusa do sljedećeg pozivanja iste funkcije (ili pozivanja *digitalRead()* ili *digitalWrite()* na istom pinu). Većina Arduina sa (sa ATmega168 ili ATmega328) kao i Nano posjeduje 6 mogućih PWM pinova (3, 5, 6, 9, 10 i 11).

3. IZRADA PROJEKTA

3.1 Praktični rad

Sveukupni fizički dio ovog diplomskog rada sastoji se od više dijelova: kutije za fotografiranje, dva bočna softboks svjetla, jedno prednje usmjereno svjetlo te difuzora i ostalih dodataka za fotografiranje te potrebne elektronike. Neki su dijelovi nastali kao vlastite ideje, a neki kao inspiracija sa interneta. Cijela kutija i rasvjeta izrađena je od bijelih materijala kao izbor vlastitog dizajna. Ovakva ideja samoupravljačke kutije za fotografiranje upravljiva pomoću Arduino mikroupravljača nije pronađena na internetu te je možda jedinstveni proizvod. Izrada kutije i ostalih dijelova zahtjeva preciznost, strpljenje, različite ideje za potrebna rješenja te eksperimentiranje sa idejama i dodane promjene i poboljšanja.

Glavni dio je kutija za fotografiranje. Donji dio napravljen je od bijele iverice. Na postolju koje je predviđeno za boce kroz sredinu je izrezan krug promjera 30 cm. Postolje je dimenzija 50x50 cm i stoji na sljepljenim i ušarafljenim daščicama također od iverice visine 12 cm koje zatvaraju rubne stranice postolja. Zadnja stranica kućišta je otvorena zbog pristupa motoru i ostalim komponentama. Rotirajući krug i ostali dijelovi od iverice koji imaju ne zaštićene rubove bez bijele površine presvučeni su bijelom plastičnom trakom. Postolje je malo povišeno kako se ne bi vidjeo pozadinski rub kruga u kadru kamere. Postolje je rotirajuće i stoji na motoru. Na donjoj strani postolja ušarafljen je plastični držač osovine motora. Valjkastog je oblika i osovina je ubodena u njega. Kako ne bi došlo do proklizavanja osovine, ušarafljena su tri vijka koja vrhovima pritišću osovinu. Osovina motora uglavljena je i na ležaj zbog smanjenja opterećenja motora. Također na rubu kruga ispod stoje tri ležaja. Oni gotovo dodiruju okruglo postolje te se postolje može osloniti. To je ostatak prvobitne ideje bez ležaja na osovini motora, samo sa tri rubna ležaja na krugu. U tom se slučaju motor uvijek opterećivao na barem jednom ležaju jer je postolje zahtjevalo uglavljenje u savršenom balansu. Rješenje samo sa ležajem na osovini pokazalo se boljim, bez opterećenja motora i odličnim za svoju svrhu. Motor je ispod postolja pričvršćen na lim koji je ušarafljen u donji dio postolja kraj izrezanog kruga.

Na drveno postolje ušarafljuje se okvirni dio kutije. Napravljen je od kutnih aluminijskih šipki koje su prefarbane u bijelu boju zbog refleksije svjetla. Kutevi su spojeni pop nitnama. Noge

okvira ušarafljene su na uglove drvenog podnožja. Sveukupna dimenzija kutije, tj. okvira od šipki je 50x50x75 cm.

Glavna rasvjeta sastoji se od dva softboksa. Softboks je kutija sa LED izvorima svjetlosti od kojih se stranice kutije šire pod nekim kutem. Dno kutije prekriveno je sa deset redova LED trake, na svakoj traci po 15 dioda. Na lijevim krajevima LED traka na žicu su zalemljeni minus krajevi, a na desnoj plus krajevi. Unutranji dio kutije prelijepljen je aluminijskom folijom zbog jače refleksije i ravnomjernog izvora svjetla. Na kraju je na kutiju naljepljen difuzor od paus papira. Dno kutije je dimenzija 25x30 cm, a vanjski dio sa difuzorom je 42x47 cm. Oba softboksa napravljena su na isti način.

Za usmjereno svjetlo (engl. *spotlight*) korištena je halogena žarulja od 55 W. Ona stoji na okruglom okviru skinutom sa ručne svjetiljke. To je pričvršćeno na jedan kraj limenog valjka koji je otvoren na oba kraja i promjera je 9 cm i dužine 25 cm. Na vanjski kraj stavljen je stožac sa odrezanim vrhom promjera 2 cm. Taj nastavak napravljen je od debljeg papira. On se može skinuti i eksperimentirati sa usmerenim svjetlom izvedenim sa kraćim/dužim stožastim nastavcima, sa užim/širim otvorom na vrhu koji je u ovom slučaju 2 cm. Na vrhu usmjerenog svjetla stavlja se čep koji na sebi ima difuzor od paus papira.

U ostale dijelove spadaju izrađeni stalci za rasvjetu, difuzori, podloga i krov kutije te ostali detalji. Sva tri svjetla imaju svoje stalke. Šipka stalka je crvena drška metle čija je dužina podesiva. Donji kraj je uglavljen u rupu na sredini podnožja koje je od smeđe dašćice. Vrh drške ima limeni držač na koji se pričvršćuje svjetlo. Cijeli softboks (i usmjereno svjetlo) moguće je usmjeriti prema gore ili dolje jer je držač pričvršćen na vijku koji omogućuje zakretanje svjetla. Difuzori su napravljeni od paus papira i od platna koje je izrezana plahta. Po dva od svakog. Paus papir stavljen je u kartonski okvir koji je moguće uglaviti u bočne stranice kutije. Drugi difuzor je plahta prikucana na letvicu na jednoj stani tako da se može spustiti po bočnoj stranici kutije. Bijeli pleksiglas je pozadina za fotografiranje koja dolazi na pozadinu kutije. Za krov kutije iskorišten je isti pleksiglas ali presvučen u crno platno jer se pokazalo da inače stvara neželjenu refleksiju odozgo kod fotografiranja staklenih boca. Sva kontrola, konekcije i elektronika je stavljena je na komad pleksiglasa koji se može pričvrstiti na zadnju stranicu kućišta podnožja. Cijela kutija je lako rastavljiva kao i rasvjeta, jer je gotovo sve spojeno vijcima.

3.2. Shema i elektronika

Za izradu upravljačkog dijela rada i realizaciju upravljanja te napajanje sustava potrebne su razne komponente i dijelovi. Korišteni su Arduino Nano mikroupravljač, koračni motor ROB-08420, motor drajver dual H-BRIDGE L298, relej modul 1-kanalni HL-51 V1.0, 3528 LED traka, halogena žarulja H3, 55W, 12V, dva NPN tranzistora TIP122, dva $1k\Omega$ otpornika, tri $10k\Omega$ otpornika, tri mikro tipkala, tri $10k\Omega$ linearna potenciometa, adapter 12V, 2A, napajanje od računala, dvije eksperimentalne pločice te skakač kablovi. Ukratko, Arduino Nano mikroupravljač upravlja cijelim sustavom. Motor drajver kontrolira pokretanje, brzinu i smjer motora na rotirajućem postolju. Komande su tipkalo za pokretanje i zaustavljanje, tipkalo za smjer motora te potenciometar za brzinu vrtnje. Usmjereno svjetlo preko releja može se paliti i gasiti na tipkalu. Intenzitet softboks rasvjete kontrolira se također pomoću potenciometa, za svaki softboks po jedan.

Blok shema je izrađena u Fritzing softveru. Cijeli sustav se napaja preko adaptera od 12V, 2A. Zbog nedovoljne struje usmjereno svjetlo ima zasebno napajanje od računala gdje se koristi jedan od njegovih 12V napajanja. Prema shemi u prilogu A.1. Arduino se napaja sa 12V iz vanjskog napajanja preko pina VIN. Arduino ima ugrađen regulator napona na 5V (pin 5V) i na 3.3V (pin 3V3). Iz tih 5V napaja se logika drivera (VLS), a sa vanjskih 12V (istih kao i Arduino) napaja se pogonskih 12V za motor. Na 5V spojeni su potenciometri i tipkala.

Tipkala S1, S2 i S3 spojeni su preko (*pull-down*) $10k\Omega$ otpornika R1, R2, R3 i oni su neka stalna vrijednost spojena preko uzemljenja i pina. To daje stanje 0 (*LOW*) kada je neaktivno, odnosno 1 (*HIGH*) kada je tipkalo aktivirano. Tipkalo S2 spojeno na pin D6 služi za pokretanje i zaustavljanje motora. Tipkalo S3 spojeno na pin D11 služi za promjenu smjera vrtnje motora. Tipkalo S1 spojeno na pin D2 pali i gasi usmjereno svjetlo preko releja U5. Potenciometri U1 i U2 spojeni na pinove A0 i A1 služe za promjenu intenziteta glavne rasvjete. Potenciometar U3 spojen na pin A2 kontrolira brzinu vrtnje motora.

Motor se kontrolira preko drajvera za motor čija su četiri izlaza spojena na motor, a četiri ulaza na Arduino digitalne pinove D7 do D10. *H-Bridge* drajveri obično se koriste za kontrolu brzine i smjera motora. Ovaj koračni motor ROB-08420 ima 200 koraka po okretaju što znači da jedan korak iznosi 1,8 stupnjeva i može napraviti 150 okretaja u minuti. Nazivni napon mu je 15.4V,

a nazivna struja 0.28A. Napaja se sa 12V. Statički moment iznosi 240 mNm. takav bipolarni motor ima dva namotaja i dvije žice za svaki. Za točnu kontrolu motora bitno je odrediti značenje žica (A,B,C,D). Prema shemi A.1. zelena je A, Crvena C, žuta B i plava D.

Prema slici 3.1. iz [8], slijedi popis konekcija na drajveru.

1 i 2 su A+ (A) i A- (C) izlazi koračnog motora.

13 i 14 su B+ (B) i B- (D) izlazi koračnog motora.

3 je 12V premosnica koja se skida ako zatreba napajanje veće od 12V DC (maksimalno 35V).

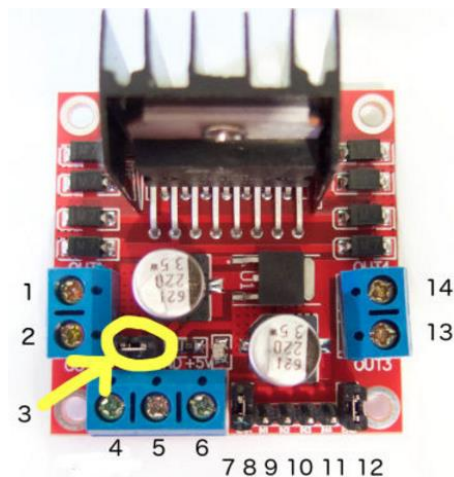
Na 4 se motor napaja. Ako je potrebno napajanje veće od 12V potrebno je maknuti premosnicu, a maksimum je 35V DC.

5 je uzemljenje (GND).

6 je napajanje od 5V ukoliko se premosnica ne ukloni.

7 i 12 premosnice ne ukanjamo ako koristimo koračni motor. Spaja se na PWM izlaz i služi za kontrolu brzine vrtnje DC motora.

8 do 11 su ulazi.



Sl. 3.1. Konekcije na dual H-BRIDGE L298 drajveru za motor

Drajver se može koristiti za više motora, a za bipolarni stepper ENA i ENB (engl. *Enable A* i *B*) moraju biti u logičkoj jedinici (+5V). Kada su CSA i CSB (engl. *Current Sense A* i *B*) u logičkoj nuli tada su takoreći onemogućeni i zato su spojeni na uzemljenje.

Modul s relejom U5 omogućava kontrolu halogene žarulje. Releji je zapravo elektronski prekidač koji se aktivira signalom kojeg pošaljemo s Arduina. Vcc se spaja na 5V, a GND se spaja na uzemljenje. In1 je izlaz spojen na Arduino pin D12. Crvena lampica na releju pali se kada je na

Vcc pin doveden napon, a zelena kada žarulja svijetli. Releji modul ima tri konekcije. NO (engl. *Normally Open*) otvoreni kontakt, COM (engl. *Common Connection*) centralna klemna i NC (engl. *Normally Closed*) zatvoreni kontakt. COM je spojen sa NC i u tom slučaju relej isključuje. Releji uključuje kada elektromagnet u releju prebaci kontakt sa COM na NO. Ako se sa pina D12 šalje stanje *HIGH* relej je uključen i žarulja svijetli. NO je spojen na plus polaritet žarulje, COM je spojen na 12V, a na NC se ne spaja ništa.

D3 i D5 su PWM izlazi iz Arduina i preko 1kΩ otpornika spojeni su na bazu tranzistora. Arduino sa PWM izlazima kontrolira tranzistore Q1 i Q2 koji rade kao sklopka te pale i gase LED-ice na određenoj frekvenciji što nije vidljivo golom oku. Na bazu dolazi struja od 5mA ($I=5V/1k\Omega=5mA$). Emiteri su spojeni na uzemljenje, a kolektori na minus polaritet svjetlećih dioda.

3.4. Programski kod

Arduino softver (*IDE - Integrated Development Environment*) moguće je besplatno preuzeti sa službene Arduino stranice. To je otvorena softver platforma pisana u programskom jeziku *Java* i bazira se na *Processing-u*. Trenutno korištena verzija je 1.6.9. Sastoji se od tekst editora za pisanje koda, područja za poruke, tekst konzole, alatne trake (brzi izbornik) s gumbima za zajedničke funkcije te niza izbornika (File, Edit, Sketch, Tools, Help). Softver se povezuje sa Arduino hardverom kako bi se na njega preneo program i s njim komunicirao. Programi napisani koristeći Arduino softver nazivaju se skice (engl. *sketches*). Skice su napisane u tekst editoru i spremljene kao datoteka s nastavkom „.ino“. Editor ima mogućnosti rezanja/lijepljenja i traženja/zamjene teksta. Područje s porukama daje povratne informacije i prikazuje pogreške. Konzola prikazuje neke osnovne podatke o skici programa uključujući potpune poruke o pogreškama i ostale informacije. U donjem desnom kutu prozora prikazan je konfiguracija pločice i serijski port (COM). Gumbi alatne trake omogućuju provjeru i prijenos programa, stvaranje i otvaranje skica programa i otvaranje serijskog monitora.

Prilikom prvog pokretanja Arduino softvera postavke su tvornički zadane i potrebno ih je definirati s obzirom na Arduino pločicu koja se koristi. U ovom slučaju koristi se Arduino Nano sa Atmega328 mikroupravljačem i komunikacijskim portom COM4. Te postavke potrebno je namjestiti u *Tools* izborniku. Nakon što je kod napisan kompajliramo ga (*Verify* okrugli gumb) zbog

eventualnih grešaka i prenosimo (*Upload* okrugli gumb) na Arduino pločicu putem USB kabela. Arduino pločica će se automatski resetirati i primiti program. Zbog korištenja nebrandirane kopije Arduina Nano sa Windowsima 7 potrebno je instalirati određeni driver zbog komunikacije. Uz potrebnu prilagodbu koda za pisanja programa korištene su i gotove skice koje su ponuđene u *File - Examples* izborniku.

Arduino programiranje uvelike je olakšano ponuđenim kodovima u softveru. Iako je bilo nekoliko problema kao što je napon za stabilno kontroliranje LED rasvjete potencijetrom i stabilizacija odskočnog vremena na tipkalima, kod je uspješno napisan. Pojedine linije koda u prilogu A.2. opisane su komentarima. Na samom početku programa uključena je Arduino knjižnica za koračni motor (*Stepper.h*). Zatim je određena revolucija motora, broj koraka motora u punom okretu od 360 stupnjeva. Motor ima 200 koraka jer je jedan korak 1,8 stupnjeva. Motor je inicijaliziran iz uključene knjižnice na pinovima 7 do 10 i nazvan je *myStepper*. Slijedi deklaracija varijabli i konstanti. LED i njihovi potencijetri postavljeni su u početnu vrijednost 0. Definirani su pinovi za sva tri tipkala i njihove varijable stanja.

Zatim su inicijalizirane *long* varijable koje prate odskočno vrijeme (engl. *bounce time*). Mjerene su u milisekundama i broj može postati velik, stoga su tipa *long*. Odskočno vrijeme se pojavljuje kod tipkala. Katkad, ako je tipkalo pritisnuto jednom može biti registrirano više uzastopnih pritisaka. Takvo sumnjivo ponašanje posljedica je mehaničkog i fizičkog problema tipkala. Potrebno je primjeniti *debounce*, naštímavanje odskočnog vremena. Potrebno je zapisati stanje promjene i ignorirati nadolazeći input za određen broj milisekundi dok se ne postigne zadovoljavajuće korištenje tipkala. Vrijeme kad je tipkalo zadnji put pritisnuto je *lastDebounceTime* (iznosi 0), a *debounceDelay* je vrijeme odgađanja za stabilizaciju signala (iznosi 50 ms).

Funkcija *setup()* poziva se kod započinjanja programa. Koristi se za inicijalizaciju varijabli i pinova. Sa *pinMode()* konfigurira se određeni pin kao ulaz ili izlaz. LED svjetala, LED lampice na Arduino i relej su izlazi, a tipkala su izlazi. Još se i sa *digitalWrite()* postavlja početno stanje releja.

Nakon kreiranja funkcije *setup()* slijedi funkcija *loop()* koja se precizno i uzastopno izvršava (vrti) kako bi program reagirao i mijenjao se sve dok se sustav ne isključi. Koristi se za aktivno upravljanje Arduino pločice. Za čitanje razine napona s potencijetra A2 (za kontrolu brzine vrtnje) koristi se funkcija *analogRead()* koja čita vrijednost iz određenog analognog pina. To je 10-

bitni analogno-digitalni pretvarač što znači da ulazni napon između 0 i 5V pretvara u *integer* vrijednosti u rasponu 0 do 1023. Sljedeća linija koda (53) je mapiranje razine napona 0 do 5V (0 do 1023) u raspon brzine vrtnje motora 0 do 150 RPM maksimalno funkcijom *map()*. Slijedi *if* petlja u kojoj se, ako je brzina motora veća od nule, u *myStepper* postavlja brzina vrtnje između 1 i 150 okretaja u minuti. Zatim se napon sa potenciometara za dvije glavne LED rasvjete (A0 i A1) također pretvara u raspon 0 do 1023 i sprema u *pot1* i *pot2*. U sljedećoj *if* petlji (linija koda 61) vrijednost 21 eksperimentalno je određena. Ustanovljeno je da napon veći od 0,1V daje nestabilan izlaz, tj. nestabilno ponašanje LED rasvjete kontroliranjem sa potenciometra. Vrijednost 0,1V predstavlja granicu stabilnog ponašanja, a dobivena je na način da se napon od 5V podijeli na 1024 i pomnoži sa 21. U petlji se vrijednost sa *pot1* također mapira u raspon 0 do 255 i sprema u *led1*. Ako je *pot1* manji od 0,1V postavlja se u nulu kako se ne bi mapirao u neku vrijednost. U izlaz *led1* pošalje se vrijednost nula i LED se ugasi. Petlje za jedan potenciometar i LED svijetlo jednake su kao i za druge (*pot2*, *led2*). Na kraju *loop* funkcije poziva se funkcija *smjerMotora()* koja određuje smjer vrtnje motora i funkcija *zaustavi()* za zaustavljanje i pokretanje motora. Zatim se D3 i D5 izlazi za LED rasvjetu definiraju kao PWM izlazi pomoću kojih je moguće mijenjati intenzitet glavne rasvjete. Više o PWM napisano je u poglavlju 2.3.

Slijedi kod za stabilizaciju odskočnog vremena (*debounce*) tipkala za gašenje i paljenje usmjerenog svjetla preko releja. Funkcijom *digitalRead()* čita se stanje tipkala (*buttonPin* - D2) u varijablu *reading* tipa *int* (vraća *HIGH* ili *LOW*). Ako je *reading* varijabla različita od zadnjeg stanja resetira se *debounce* tajmer. Pomoću funkcije *millis()* broji se vrijeme rada programa i broj u milisekundama upisuje se u varijablu *lastDebounceTime*. Zatim se očitava novi, trenutni broj milisekundi i oduzima broj upisan u *lastDebounceTime*. Ako je ta razlika veća od *debounceDelay* znači da je prošlo više od vremena potrebnog za stabilizaciju odskočnog vremena i trenutno stanje tipkala se može uzeti za stvarno stanje. Prema *if* petlji na liniji koda 107, ako je trenutno očitavanje (*reading*) različito od zadnjeg upisanog trenutnog stanja kao trenutno stanje upisuje se ono što je trenutno i očitano. Zatim se u sljedećoj petlji mijenja stanje izlaza samo ako je stanje tipkala vrijednost *HIGH* (stisnuto tipkalo). Nakon toga se svjetlo pali ili gasi preko releja te se trenutno očitavanje (*reading*) upisuje u *lastButtonState*. Kada program sljedeći put prođe kroz petlju ovo trenutno stanje bit će prošlo stanje.

Slijedi gotovo isti kod, ali za stabilizaciju odskočnog vremena tipkala za smjer motora. razlika je u *if* petlji za vrijeme mirovanja motora. Globalna varijabla *stanje* postavljena je u *HIGH*, što predstavlja mirovanje motora (motor pravi nula koraka). Motor se vrti kada je ona *LOW*. Ako je varijabla *smjer* motora u *HIGH* korak ima vrijednost „1“ i motor se vrti u pozitivnom smjeru, smjeru kazaljke na satu. U tom se slučaju na Arduinu pali LED lampica „L“ koja je na pinu D13. Kad se motor vrti u negativnom smjeru korak je „-1“ i LED se gasi. Posljednji dio koda također se odnosi na stabilizaciju odskočnog vremena tipkala za pokretanje i zaustavljanje motora.

3.5. Moguća poboljšanja

Kao mane ili moguća poboljšanja gotovog proizvoda su jačina rasvjete, dimenzije softboksa i povišenost postolja. Kao glavna rasvjeta korištena su dva softboksa i u svakom je 150 svjetlećih dioda tipa 3528. Dobro rješenje bile bi svjetleće diode 5050 koje daju 16 do 22 lumena svjetlosnog toka, dok korištene daju 6 lumena po diodi. 5050 LED se zapravo sastoji od seta od tri 3528 svjetlećih dioda unutar jedne. Jača rasvjeta rezultirala bi jasnijim odsjajem. Nešto su veće i malo skuplje. Softboksi služe kao rasvjeta koja na bočnim rubovima boce na fotografiji daje bijele odsjaje. Bolje rješenje bio bi nešto už i višlji okvir softboksa jer bi se u tom slučaju dobila uža i jasnija crta odsjaja. Kako bi dobili už softboks moguće je rubove prekriti sa nepropuštajućim površinama ili ga zarotirati ukoso kako ne bi bio paralelan sa stranicom kutije. Boca za fotografiranje nalazi se na sredini okruglog postolja. Prednje usmjereno svjetlo postavljano je ispod razine dna kutije usmjereno ukoso odozdo prema etiketi boce jer se na taj način dobilo najbolje osvjetljenje. Prednji rub dna kutije zaklanjao je svjetlosni snop usmjerenog svjetla te bi nešto višlje rotirajuće postolje bilo bolje rješenje. Još jedno poboljšanje bilo mi izrada još nekoliko izvora usmjerenih svjetala kako bi sjajne etikete zbog svoje zaobljenosti bile jednakomjerno osvjetljene.

Korak motora od 1,8 stupnjeva pokazao se praktičnim i preciznim za željenu rotaciju položaja boce u kadru. Problem se javljao kod tipkala za zaustavljanje rotacije. Zbog osjetljivosti na odskočno vrijeme pritiska tipkala malo je teže procijeniti točan trenutak željenog zaustavljanja motora. To se odnosi na nešto veće brzine od minimalne. Moguće rješenje je korištenje najmanje brzine vrtnje kod preciznog okretanja. Još bolje rješenje bio bi npr. program u kojem postoje određene vremenske pauze rotacije između dva koraka pomaka. Rotacija pri većim brzinama korisna je kod snimanja promotivnih videa ili animacija gdje se boca elegantno okreće.

4. FOTOGRAFIRANJE STAKLENIH BOCA I ČAŠA PIĆA

4.1. Gradijent refleksija sa velikom softboksom

Fotografiranje sjajnih predmeta, naravno razlikuje se od portreta. Za portrete bi koristili veliki izvor svjetla koji je teško kontrolirati na predmetima manjih dimenzija. Jedan od najtežih aspekata je dobivanja točno željene difuzije kada je u pitanju korištenje softboksa. Kada se primjenjuje pogrešna količina difuzije ili je ispravna količina pogrešno primjenjena može doći do oštre refleksije i minimalnog gradijenta, što obično nije rezultat koji pokušavamo postići. Gradijent predstavlja usmjerenu promjenu intenziteta ili boje na fotografiji. Zapravo je postupno mješanje ili prijelaz nijansi, tj progresija boje.

Sljedeći slučaj odnosi se na na jednostavan primjer sjajne crne šalice sa žlicom unutra. Izvor svjetla je sa lijeve strane šalice, a drška šalice na desnoj strani. Šalicu je dobro smjestiti na manju površinu u zrak kako nikakva površina ispod nje ne bi uzrokovala odsjaj. Ukoliko na šalici dobijemo odsjaj kojim nismo zadovoljni možemo koristiti gradijent. Cilj je dobiti gladak prijelaz kako bi se što vjernije prikazao oblik šalice. To se može postići tako što ispred softboksa postavimo okvir sa difuzorom. Ako pomičemo razmak između softboksa i difuzora ili čak mijenjamo jakost svjetla nećemo dobit neke velike promjene, stoga je bitno zarotirati softboks tako da se na jednom rubu dodiruje sa difuzorom u obliku slova „V“. Cilj je prikazati sjajnost predmeta, a time postizemo oštru promjenu odsjaja na šalici i dobivamo vjerniji prikaz zaobljenosti. Greška koja se pojavljuje kod fotografiranja je da sjajan predmet izgleda kao da je mat boje. Kako bi dobili bolji gradijent sljedeća ideja je sužavanje površine softboksa. Dovoljno je s jedne strane softboks prekriti površinu neprozitnim materijalom dovoljnih dimenzija da prekrije željenu dio. Najbolje je isprobati slikanjem i usporedbom fotografija sa prekrivanjem većine površine softboksa, znači uskog izvora svjetla, do sve šireg. Prije toga možemo pojačati izvor svjetla zbog boljeg učinka jer smanjujemo površinu izvora. Za još bolji efekt sa desne strane šalice možemo dodati reflektirajuću površinu paralelno sa suprotnim difuzorom i jedan izvor svjetla iza šalice, više prema desno koji je ukoso usmjeren prema šalici. To nam sada daje najbolju predodžbu koliko je zapravo predmet koji slikamo sjajan. Dobit ćemo oštri osvijetljeni rub na desnoj strani šalice. Za što bolje željene rezultate možemo utjecati na promjenu odsjaja na šalici eksperimentiranjem sa jačinom izvora svjetla iza šalice, njegovim usmjeravanjem, zatim kutom nagiba reflektirajuće podloge desno ili njene udaljenosti. Rotiranjem

reflektirajuće podloge kontroliramo širinu neobasjanog dijela šalice koji se nalazi na prednjoj strani šalice i izgleda kao tamnija granica između lijevog i desnog gradijentiranja. Cijela ideja ovoga je prikazivanje odsjaja nekog sjajnog predmeta. Može se iskoristiti i za mnogo sjajnije predmete kao što su nakit, satovi, metal itd.

4.2. Kvalitetno fotografiranje proizvoda

U nastavku je obrazloženo fotografiranje boca i čaša ispunjenim tekućinom tj. pićem. Postava za slikanje zna biti zahtjevna kada je u pitanju fotografiranje stakla i tekućina. U pitanju je fotografiranje artikala, reklamnih sadržaja, u ovom slučaju boca viskija sa čašom u kojoj je natočena ista tekućina i kockama leda. Profesionalni fotografi u ovakvim slučajevima koriste lažne akrilne kockice. Za podlogu na kojoj stoje predmeti najbolje je koristiti crnu sjajnu površinu kakva se koristi u ovakvoj vrsti fotografiranja. To može biti pleksiglas ili crno staklo i slični materijali čiji odsjaj daje jedan elegantan dojam i izgled proizvoda. Sa lijeve strane, bliže kameri, imamo spotlight svjetlo usmjereno na etiketu boce. Iza boce je zanimljivo napraviti efekt svjetla koje se polukružno penje, tj. isijava po podlozi odozdo. To možemo postići tako da na udaljenu podlogu postavimo reflektor usmjeren prema donjem dijelu te podloge.

Cilj fotografije je dobiti dubinu boce u dalj od kamere, oštrinu boce, dok je čaša izvan fokusa. Odmah iza boce i čaše postavljaju se mali uspravljani paneli sa reflektirajućom površinom izrezani u obliku predmeta u visini razine tekućine da se ne bi vidjeli u kadru jer su zaklonjeni. Oni daju bogati sjaj koji oživljava proizvod. Blago su ukošeni prema glavnom izvoru svjetla koji se nalazi sa desne strane. Izvor svjetla su dva uska softboksa iza velikog difuzora. Udaljeniji softboks je primaknut bliže difuzoru a drugi malo udaljen. Iza udaljenijeg postavljena je crna podloga kako svjetlo ne bi osvjetljavalo pozadinu fotografije jer je pozadina već osvjetljena kao što je to spomenuto. Razlog zašto su dva softboksa je taj što jednim želimo postići cjelokupnu rasvjetu, a drugim rubnu liniju po dužini boce.

Ako bi maknuli difuzor desila bi se glavna greška koju fotografi rade kada osvjetljuju boce za fotografiranje jer se dobije grubi blok svjetlosti na boci. Zato difuzorom postizemo gradaciju odsjaja, što je efekt koji želimo postići. Ako stavimo jedan uski softboks, on stvara mekano osvjetljeni rub (desni) koji odvaja oblik boce od pozadine. Udaljenost svjetla od difuzora utječe na istaknutost odsjaja ruba boce. Što je udaljeniji izvor, slabiji je odsjaj. Osim udaljenosti također

utječe i položaj izvora svjetla ako ga pomicemo po okomitoj liniji jer svjetlost drugačije obihvaća zakrivljenu površinu boce. Na taj način kontroliramo rubni odsjaj. Drugi softboks koji se nalazi bliže kameri služi kao glavna rasvjeta i zato je malo udaljeniji da svjetlo ne bude prenaplašeno. Također su bitni i mali reflektirajući paneli iza boce i čaše. Uloga im je reflektirati svjetlo kroz bocu ali da nisu vidljivi u kadru, te treba paziti na oblik u kojem se izrezuju. To bi bila cijela postava ovog načina fotografiranja, fizike svjetla pomješane sa malo umjetnosti. Svladavanjem problema osvjetljenja kreativnim rješenjima mogu se postići odlični rezultati.



Sl. 4.1. *Primjer finalne fotografije opisanog slučaja boce i čaše viskija*

4.3. Osvjetljavanje čaše vina (slikanje posljednje kapljice)

U nastavku je opisan je jednostavan primjer fotografiranja je i gdje u čašu crnog vina ulijevamo zadnje kapljice vina. Crnim panelima zatvorimo lijevu i desnu stranu prostora u kojem stoji čaša. Podloga je također crna. U čašu ulijemo vino, ne previše. Izvor svjetla postavimo iznad. U obliku polukruga izrežemo reflektirajuću foliju i pričvrstimo ju na komadić ravne žice. Otprilike oblika čaše do razine tekućine. To postavimo iza čaše tako da žica prolazi iza stalka čaše a folija iza natočenog vina tako da se ne vide na fotografiji. Foliju uvijek možemo malo nagnuti prema nazad, tj. koso usmjeriti prema izvoru svjetla jer nam je cilj osvijetliti površinu vina u čaši tako da se dobije polukružni bijeli, osvijetljeni rub na zadnjoj strani razine vina. Također se na površini vina jasno vidi

crvena boja dok je ostali dio tekućine ispod crn. S lijeve strane postavljamo komad reflektirajuće folije također ukoso prema gore, a desno ravan bijeli panel. Tako dobijemo blagi odsjaj ruba čaše koji je nešto istaknutiji s lijeve strane gdje je folija. Dobijemo jasan prikaz oblika čaše. Sada možemo uliti nekoliko kapljica vina u čašu iz boce koju polako odmičemo i napraviti animaciju ulijevanja zadnje kapljice ili fotografiju sudaranja kapljice sa površinom vina. Zbog dobrog rasporeda opisanog osvjetljenja, boca također bude pravilno osvjetljena.



Sl. 4.2. *Primjer finalne fotografije opisanog slučaja čaše crnog vina i zadnje ulijevane kapljice*

4.4. Fotografiranje boca i pića

Nešto drugačija tehnika osvjetljivanja kod fotografiranja boca s pićem je propuštanje svjetla na stražnji dio boce kroz pozadinu izrezanu u obliku boce. Pozadina može biti nekakav papir koji ne propušta svjetlo, željene boje podloge. Ako kameru ostavimo stacionarno na jednom mjestu, netko nam može pomoći obilježiti oblik boce na udaljenoj podlozi kako bi to mogli izrezati. Rub se naravno ne smije vidjeti, a izrez ne mora biti točan nego što bliži obliku boce. Bit će izrezano u većem mjerilu jer je pozadina udaljenija od boce. Jedan izvor svjetlosti postaviti ćemo iza podloge uperen u izrez prema boci. Kamera će vidjeti bocu i svjetlost propuštenu kroz nju. Nakon što ugasimo svjetla u prostoriji vidjet ćemo da osvjetljenost boce ovisi o poziciji izvora svjetla. Da li je ono postavljeno niže ili višlje u odnosu na razinu boce. Ukoliko pozadina propušta pozadinsku svjetlost kroz ne izrezani dio, tada se iza te podloge može staviti nešto ne propuštajuće također izrezano identično. Zgodan detalj je kraj boce postaviti voće od kojeg je piće proizvedeno. U ovom slučaju recimo da je to jabuka. Tada i jabuku želimo osvjetliti pa se s jedne strane postavlja veliki softboks a na suprotnu stranu reflektirajući materijal. Tada se javlja problem što kroz bocu prolazi

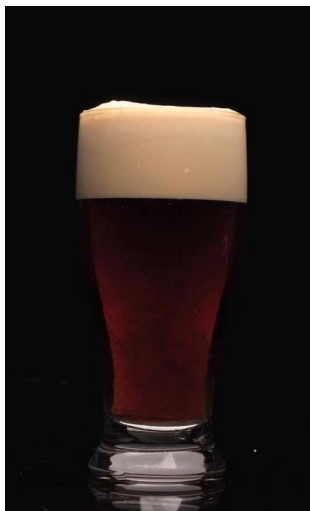
previše svjetla jer je sada obasjana i bočno. Potrebno je pronaći balans kako bi dobili željene rezultate. To možemo postići mijenjajući jakost izvora svjetlosti.



Sl. 4.3. Primjer finalne fotografije boce slikane sa izrezanom pozadinom

4.5. Fotografiranje čaše i boce piva

Jednostavan primjer fotografiranja čaše piva sa reflektirajućim panelom sastoji se od svjetla iznad čaše, crne podloge i pozadine. Nakon što u čašu ulijemo pivo do željene razine vidjet ćemo da je sadržaj taman. To možemo riješiti postavljanjem reflektirajućeg panela izrezanog u obliku čaše do razine tekućine. Zanimljiv detalj je stvaranje efekta kapljica po čaši. To možemo postići sprejanjem vode iz bočice te dobijemo efekt kondenzacije. Voda se nanese samo do razine tekućine jer pjena iznad nije hladna i ne stvara kondenzaciju na staklu pa se taj dio može prekriti tijekom sprejanja. Dok sve to namjestimo pjena piva se može spustiti, ali i za to postoji rješenje. Dovoljno je u pivo staviti malo soli i pjena će se izdići. Ukoliko to ne djeluje onako kako smo zamislili uvijek možemo uliti još malo piva.



Sl. 4.4. *Primjer finalne fotografije čaše piva*

Za fotografiranje boce piva osim klasičnih izvora svjetlosti mogu se koristiti i blicevi. Prednost bliceva je što su uski i puno manji od softboksa. Te ih možemo sakriti iza boce i lako ih usmjeravati. Više njih može se postaviti na manjem području što je nemoguće sa nekoliko softboksa. Jedan blic se postavi iza boce, gotovo na dno i šalje svjetlost kroz bocu prema kameri. Namjesti se na najmanju snagu bljeska. Na bočne strane te iznad boce postavimo tri disk bljeskalice (engl. *flash disc*). Disk bljeskalica je blic okruglog oblika promjera 30cm koja se može namjestiti da okida u trenutku nakon što detektira drugi bljesak (engl. *slave mode*), u ovom slučaju onaj iza boce koji ručno okidamo prekidačem kada slikamo. Disk bljeskalica iznad boce osvjetljava grlo boce i čep, te ako bi koristili samo tu disk bljeskalicu iznad i blic iza boce, vidjeli bi blago osvjetljen vanjski oblik boce i svjetlo koje prolazi kroz bocu, što je dobar početak. Kod fotografiranja boca bitno je prikazati liniju „ramena“ boce te ju tako istaknuti na pozadini. Za to služe bočne disk bljeskalice, obje blago nagnute prema dolje i prema stražnjem dijelu boce. Sada bi to sve odlično osvjetlilo bocu ali bez osvjetljenja prednje etikete. Naravno, veoma je bitno vidjeti ime proizvođača, pa za to možemo koristiti svjetlo usmjereno od kamere, malo iznad boce da se ne vidi u kadru. To svjetlo ne smije biti prejak te se može prigušiti rešetkom koja se postavlja preko svjetla. Npr. rešetka koja je zapravo guma sa izbušenim rupicama. Sva svjetla dosad spomenuta odlična su rasvjeta, ali za ovakvo fotografiranje potrebni su detalji kako bi dobili savršenu krajnju fotografiju. Dodaje se umjetna kondenzacija vode na boci i led. Na površinu (najbolje crna reflektirajuća) možemo staviti kapljice vode i sitne komadiće leda. Led se brzo topi stoga fotografi koji se bave ovakvim fotografiranjem koriste lažni led. Nakon što je fotografiranje sa pravim ledom obavljeno i

on se otopi. Poslije toga se možemo fokusirati na fotografiranje detalja na boci i sve to kasnije spojiti na računalu. Kapljice vode na boci možemo nanijeti bočicom za prskanje. Najbolje je koristiti glicerín pomiješan sa vodom, što je zapravo trik da kapljice na boci ostanu dovoljno dugo.



Sl. 4.5. *Primjeri finalne fotografije boce piva sa umjetnim efektima*

4.6. Efekt hladne boce

Postoji nekoliko slojeva koji se nanose kako bi dobili izgled hladne boce. Kako bi stvorili izgled zaleđenosti ili umjetni mraz na bocu nanosimo vosak. Nakon što ga naneseemo na mjesto koje želimo utrljamo da rukom da izgleda vjernije. Tada izgled kao da je boca upravo izvađena iz hladnjaka.

Sljedeći korak je napraviti kapljice na boci. Za to se koriste male bočice za sprejeve čija veličina zapravo ovisi o veličini kapljica na boci. Za to je dobar kukuruzni sirup pomiješan sa vodom u omjeru 1:5. Nakon što voda ispari kukuruzni sirup ostaje na boci i izgleda kao voda. Ako sprej više približimo boci dobit ćemo veće kapljice, a ako udaljimo, manje. Zanimljiv detalj su kapljice koje se cijede niz bocu i ostavljaju trag. Za tu svrhu postoji prozirni gel koji je komercijalni proizvod pod nazivom „*aquagel*“. Taj gel je gust i ne ciedi se pa se na bocu nanosi tako što se vrh uskog štapića umoči u gel i nacrtá na boci. Kreće se odozdo prema gore i crta se trag po kojem bi inače curila kapljica, koja ne mora nužno ostavljati ravan trag. Najvjerniji se efekt dobije ako ga naneseemo na bočne strane boce ili na prednji dio.

Sljedeći trik je napraviti da čaša za pivo ima ledeni izgled. Postoji sprej baš za takvu upotrebu napravljen za potografe i zove se „*Dulling Spray*“. Služi kako bi se privremeno smanjio

odsaj predmeta koji se snimaju i daje mutan izgled. Prije nego čašu nasprejamo, na godnji dio možemo nalijepiti traku da bi zaštitili dio čaše u kojem bi bila pjena piva. Sprej je dobro prije korištenja staviti u toplu vodu kako bi se sadržaj bolje rasporedio. Sprej se nanosi tako da istovremeno mičemo i čašu i sprej dok sprejamo kako bi se dobio jednolik izgled. Sprej nije dobro previše približiti čaši. Nakon toga na čašu možemo nanijeti glicerini pomješani sa vodom u omjeru 50:50 kako bi dobili kapljice. Zatim na čašu i na bocu možemo staviti usitnjene komadiće leda koji postoje i kao komercijalni proizvod pod nazivom kristalni led. Cilj je nanijeti malo zbog efekta zaleđenosti i ne dirati površinu čaše koju smo nasprejali prije toga. Pomoću ovih tehnika dobije se efekt hladnih boca pića i čaša koji izgleda veoma prirodno.



Sl. 4.6. Primjer finalne fotografije boce piva sa umjetnom kondenzacijom

4.7. Pleksiglas za fotografiranje proizvoda

Osim samog osvjetljenja bitan faktor je i odabir površine na koju stavljamo predmete. Korisno bi bilo staklo na jednoj strani obojano u crno tako da ima refleksiju kao ogledalo samo crnu, ali u nekim slučajevima dobivamo duplu refleksiju kao što je to slučaj kod ogledala. Zato je najbolje koristiti crni pleksiglas. Dosta je skuplji u odnosu na staklo, ali za ovakvo fotografiranje ne trebamo velike površine. Prednost je što ga je recimo lakše izrezati. Mana je što se pleksiglas lako ogrebe te moramo biti pažljivi. Kod pleksiglasa možemo koristiti njegove dvije strane, dvije površine. Jednu možemo zaštititi npr. nekakvim preljepljenim slojem i koristiti ju nakon nekog vremena ukoliko izgrabamo prvu površinu. Bilo kakav predmet koji postavimo na takvu površinu izgleda dobro jer

dobiva elegantan odsjaj. Odsjaj je crn ali ga lako možemo pretvoriti u bilo koju boju koristeći pozadinu željene boje koja se reflektira od pleksiglas.

4.8. Deset bitnih stvari za dobru fotografiju vina

Nakon što je fotografiranje obavljeno, svoju finalnu fotografiju možemo provjeriti u softveru za obradu slike kao što je *Photoshop*.

1. Kako boca izgleda sa različitim pozadinama? Dobro je isprobati različite boje pozadine nakon fotografiranja. Različiti se odsjaji na staklu mogu različito doživjeti na raznim pozadinama.
2. Da li etiketa stoji ravno i u centru? To možemo provjeriti crtama vodilicama u Photoshop-u. Vodoravni rubovi etikete zaobljeni su zbog zaobljenosti boce te im lijeva i desna krajnja točka moraju biti na istoj razini.
3. Da li je logo na grlu boce centriran? Grlo boce na vinima često je prekrivno folijom na kojoj je logo. Logo također treba biti centriran i na sredini.
4. Da li ima izobličenja u objektivu? To se može vidjeti ako je boca na fotografiji blago nagnuta gledano iz horizontalne i vertikalne perspektive. Tada boca izgleda blago ukošeno kao kad fotografiju gledamo sastrane a ne ravno ispred. Isto tako leća može izobličiti bocu tako da izgleda izbočeno ili uvučeno na rubovima gdje je zapravo ravna. Također se nekad može vidjeti i tanka crta npr. žute, zelene ili plave boje koja se neželjeno pojavljuje najčešće na rubovima etikete.
5. Da li se vidi spoj stakla na boci ili grlu? To je crta gdje je spojeno staklo kad je boca ljevana u proizvodnji. Bitno je ukloniti vidljivi spoj pogotovo ako se pojavljuje na osvijetljenom dijelu gdje je najviše vidljiv.
6. Da li osvijetljenost izgleda realistično i trodimenzionalno? Ukoliko krivo postavimo rasvjetu, dobit ćemo ne realistične odsjaje na boci koji moraju imati smisla.
7. Da li su ruboci boce glatki? Boce se osvijetljavaju tako da se na bočnim rubovima dobije bijeli odsjaj. Zato je dobro staviti crnu podlogu i provjeriti ravninu rubova boce.
8. Da li na boci ima prašine ili dovoljna količina prašine? To je vidljivo samo ako se boca dovoljno zumira. Malo prašine na boci može dati realističan prikaz, ali previše prašine može dati dojam da je boca prljava.
9. Da li postoji neželjeni oblik kalupa na boci? Često na donjem dijelu boce kod dna postoje npr. točkice na staklu ili sitni izbočeni tekst.

10. Da li su boje vjerno prikazane? To je teško odrediti jer svaki monitor drugačije prikazuje boje. Zato je dobro pobrinuti se da fotografiramo sirovo bez procesiranja podataka tokom slikanja. Također je dobro koristiti sivu karticu pomoću koje radimo balans bijele boje. Siva kartica ima 18% sive boje i fotografira se u okruženju koje slikamo te služi kao vodič za prikaz boja.

4.9. Korištenje kutije za fotografiranje

Kutija sa svim dijelovima je praktična i dobro služi svojoj svrsi za koju je namjenjena. U nastavku su priložene fotografije različitih boca sa eksperimentalnom postavom rasvjete. Na slici 4.7. vidljiv je izgled konačnog izgleda proizvoda. Slika predstavlja namještanje postave rasvjete za fotografiranje boce vina.



Sl. 4.7. *Namještanje postave rasvjete za fotografiranje*

Na slici 4.8. vidljivo je kako usmjereno svjetlo jednakomjerno obasjava etiketu boce vina. Na ovoj fotografiji upaljen je samo desni softboks sa maksimalnim intenzitetom svjetla. Zbog toga je vidljiv samo desni odsjaj na boci.



Sl. 4.8. Razlika fotografije bez (lijevo) i sa (desno) osvjetljenjem etikete usmjerenim svjetlom

Slika 4.9. lijevo prikazuje crnu bocu vina koja je praktičnija za fotografiranje zbog neprozirnosti stakla. Na crnom staklu se jasnije vidi odsjaj i gradijent odsjaja. Grlo boce prekriveno je sjanom folijom koja također odlično prikazuje gradijent odsjaja. U ovom je slučaju lijevi softboks namješten da svijetli slabije od lijevog. Slika 4.9. desno daje najbolji rezultat zbog drugačijeg rasporeda rasvjete. Oba softboksa svijetle najvećim intenzitetom i etiketa je jasnije osvjetljena zbog različitog kuta položaja usmjerenog svjetla.



Sl. 4.9. Fotografije crne boce sa dva različita postava rasvjete

Kod boce piva problem je zlatna etiketa koja zahtjeva ravnomjerno osvjetljenje sa više izvora usmjerenog svjetla. U ovom slučaju sjajna slova nisu dovoljno dobro čitka zbog promjene nijanse zlatne boje.



Sl. 4.10. *Fotografije boce piva i čaše sa naknadnim posvjetljenjem u Photoshop-u (lijevo) i originalne fotografije (desno)*

Ideja kutije i njena svrha je fotografiranje bez dodatne obrade fotografija u softveru. Slika 4.10. iznad, kao i ostale uslikane fotografije pokazuju da bi jača rasvjeta dala bolji rezultat. Najbolje je to vidljivo na tamnijoj pozadini.

5. ZAKLJUČAK

U ovom diplomskom radu napravljena je kutija za fotografiranje, dva bočna softboks svjetla, jedno prednje usmjereno svjetlo, te difuzori i ostali dodaci za fotografiranje. Arduino Nano mikroupravljač upravlja cijelim sustavom. Kako bi sustav bio upravljiv spojena je potrebna elektronika i isprogramiran mikroupravljač. Motor drajver kontrolira pokretanje, brzinu i smjer motora na rotirajućem postolju. Komande su tipkalo za pokretanje i zaustavljanje, tipkalo za smjer motora te potencijometar za brzinu vrtnje. Usmjereno svjetlo preko releja može se paliti i gasiti na tipkalu. Intenzitet glavne rasvjete kontrolira se također pomoću dva potencijometra. Sve radi ispravno i kutija sa svim dijelovima je praktična, te dobro služi svojoj svrsi za koju je namjenjena, ali nisu zadovoljena sva očekivanja. Bolja rješenja su veći i uži softboksi, jača LED rasvjeta, povišenije rotirajuće postolje i drugačije rješenje precizne rotacije postolja.

LITERATURA

- [1] Što je Arduino, 13.6.2016.
<https://hr.wikipedia.org/wiki/Arduino>
- [2] Arduino okruženje, 13.6.2016.
<https://en.wikipedia.org/wiki/Arduino>
- [3] Uvod u Arduino i zašto ga koristiti, 13.6.2016.
<https://www.arduino.cc/en/Guide/Introduction>
- [4] Croduino kao hrvatska verzija Arduina, 13.6.2016.
<https://e-radionica.com/hr/blog/2015/10/08/sto-je-arduino-i-croduino/>
- [5] Arduino Nano pregled, 13.6.2016.
<https://www.arduino.cc/en/Main/ArduinoBoardNano>
- [6] Modulacija širine impulsa (PWM) i Arduino, 20.6.2016.
<https://www.arduino.cc/en/Tutorial/PWM>
- [7] PWM i *analogWrite()* funkcija, 20.6.2016.
<https://www.arduino.cc/en/Reference/AnalogWrite>
- [8] Konekcije na drajveru za motor, 22.6.2016.
<http://www.instructables.com/id/Control-DC-and-stepper-motors-with-L298N-Dual-Moto/>

SAŽETAK

Naslov: Mikroupravljački sustav kutije za snimanje fotografija

U ovom diplomskom radu izrađen je mikroupravljački sustav kutije za fotografiranje pića u promotivne svrhe. Kutija za fotografiranje boca i čaša pića omogućava zamjenu pozadine, difuzora na stranicama i stropu. Ima rotirajuće postolje na koračnom motoru, dva glava softboks svjetla izrađena od LED trake i prednje usmjereno halogeno svjetlo za osvjetljavanje etikete. Sustav se upravlja isprogramiranim Arduino Nano mikroupravljačem. Pokretanje, smjer i brzina motora kontroliraju se preko dva tipkala i potencijometra pomoću dražveza za motor, glavna rasvjeta omogućuje promjenu intenziteta svjetla na potencijometrima i usmjereno svjetlo uključuje se preko releja sa tipkalom. To sve spojeno je na Arduino Nano pločicu. U pismenom dijelu rada opisani su načini osvjetljenja kod fotografiranja staklenih boca i čaša, trikovi i efekti za dobivanje što boljeg rezultata konačne fotografije, te je to primjenjeno kod vlastitog fotografiranja pomoću izrađene opreme. Teoretski je opisan Arduino Nano mikroupravljač, postupak izrade praktičnog rada, shema spajanja, sva elektronika, programski kod na mikrokontroleru i ostali pojmovi potrebni za izradu sustava kao što je modulacija širine impulsa. Kutija i sva oprema odlično služe svojoj namjeni, ali postoje i moguća poboljšanja koja su opisana.

Ključne riječi: koračni motor, dražver za motor, softboks, Arduino Nano, modulacija širine impulsa, difuzor, LED

ABSTRACT

Title: Microcontrolled lightbox setup for product photography

In this thesis, microcontrolled setup for commercial product photography was made. This lightbox for shooting drink bottles and glasses allows replacement of background, side diffuser panels and ceiling. It has rotating stand on the stepper motor, two main softboxes with LED stripe and front halogen spotlight for lighting bottle label. The system is operated by programmed Arduino Nano microcontroller. Start and stop, direction and speed of the motor are controlled via two push buttons and potentiometer using motor driver. Intensity of the main lightning can be changed with potentiometers and spotlight can be turned on and off using relay. Everything is connected on Arduino Nano board. Written thesis describes ways of lighting for drink photography, tricks and effects to achieve best results on the final image. That is applied for taking pictures using made equipment. Arduino Nano microcontroller, making process, electronic scheme, electronics, program code uploaded on the microcontroller and other terms required to understand and create these system such as pulse-width modulation are also described in this thesis. The lightbox and its equipment serve excellently their purpose, but there are potential improvements which are also described.

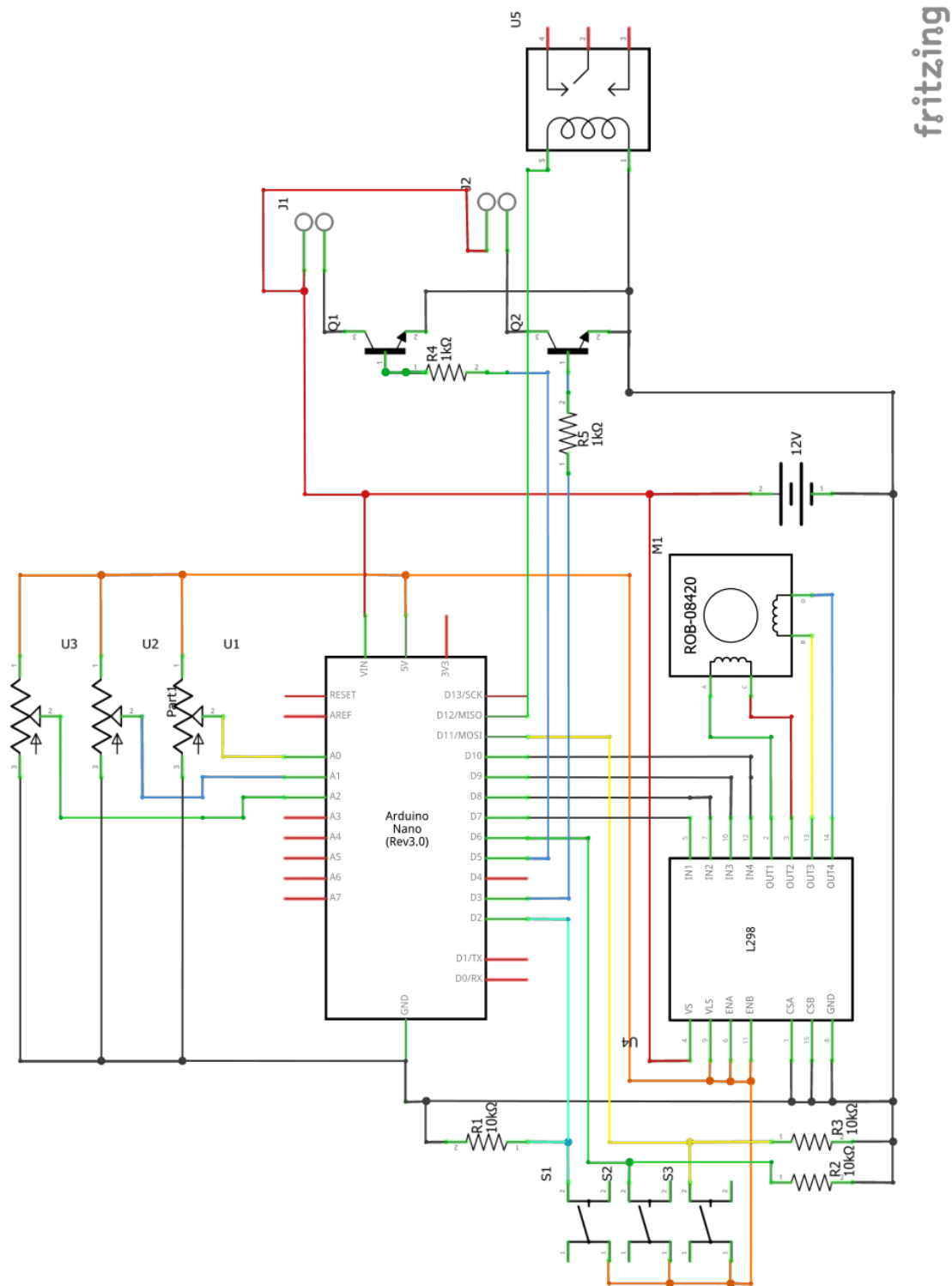
Keywords: stepper motor, motor driver, softbox, Arduino Nano, pulse-width modulation, diffuser, LED

ŽIVOTOPIS

Ivan Reponj rođen je 7. ožujka 1989. godine u Požegi. Srednju tehničku školu sa smjerom elektrotehničar završio je 2004. godine u Požegi gdje je pohađao i osnovnu školu. U srednjoj školi je sudjelovao na županijskom natjecanju iz hrvatskog jezika. Godine 2008. upisuje Elektrotehnički fakultet u Osijeku, stručni studij elektrotehnike, smjer Informatika. Nakon stručnog studija upisuje razlikovnu godinu te 2013. godine nastavlja s preddiplomskim studijem računarstva.

PRILOG A: Elektronička shema sustava

U ovom prilogu slika A.1. predstavlja elektroničku shemu sustava upravljanja. Izrađena je u Fritzing softveru.



fritzing

PRILOG B: Programski kod sustava

U ovom prilogu nalazi se programski kod sustava upravljanja napisan u Arduino softveru i pohranjen na Arduino mikroupravljač.

```
#include <Stepper.h> // uključivanje Arduino knjižnice za koračni motor
```

```
const int stepsPerRevolution = 200; // Motor ima 200 koraka u jednom okretaju (360), a jedan korak je 1,8 stupnjeva.
```

```
Stepper myStepper(stepsPerRevolution, 7, 8, 9, 10); // Motor je nazvan myStepper, i definiran je Stepper (iz biblioteke Stepper.h). Preko drivera se kontrolira na pinovima 7 do 10.
```

```
int pot1 = 0; // Potencimetri za LED postavljeni na početnu vrijednost 0.
```

```
int pot2 = 0;
```

```
int led1 = 0; // LED postavljene na početnu vrijednost 0.
```

```
int led2 = 0;
```

```
const int relej = 12; //Definiran je pin D12 na koji je spojen relej.
```

```
const int smjerPin = 11; // Definiran je pin D11 tipkala za promjenu smjera.
```

```
boolean smjer = true; // Vrijednost true predstavlja vrtnju motora u smjeru kazaljke na satu (clockwise). U tom se slučaju pali Arduino LED 13.
```

```
int smjerButtonState; // Pamti stanje smjera motora.
```

```
int lastSmjerButtonState = LOW; // Prošlo stanje tipkala za promjenu smjera. Ako je prošlo stanje jednako trenutnom ne događa se ništa (kasnije u programu).
```

```
int zaustaviPin = 6; // Definiran je pin D6 tipkala za pokretanje i zaustavljanje motora.
```

```
int stanje = HIGH; // Sve isto kao i kod promjene smjera, samo je ovdje za pokretanje i zaustavljanje.
```

```
int zaustaviButtonState; // Pamti stanje pokrenut/zaustavljen.
```

```
int lastZaustaviState = LOW; // Prošlo stanje tipkala za pokretanje i zaustavljanje motora.
```

```
const int buttonPin = 2; // Definiran je pin D2 na koji je spojeno tipkalo za kontrolu releja.
```

```
int ledState = HIGH; // HIGH stanje znači upaljeno tipkalo.
```

```
int buttonState; // Pamti stanje tipkala preko releja.
```

```
int lastButtonState = LOW; // Prošlo stanje tipkala preko releja.
```

```
// the following variables are long's because the time, measured in milliseconds,
```

```
// will quickly become a bigger number than can be stored in an int.
```

```
long lastDebounceTime = 0; // the last time the output pin was toggled
```

```
long debounceDelay = 50; // the debounce time; increase if the output flickers
```

```
long lastDebounceTime1 = 0; // Vrijeme kad je tipaklo zadnji put stisnuto
```

```
long debounceDelay1 = 50; // vrijeme potrebno za debounce (stabilizaciju) signala. Povećati ako izlaz "treperi", ako se ne ponaša dobro.
```

```
long lastDebounceTime2 = 0;
```

```
long debounceDelay2 = 50; //Ovo su tri debounce vremena za sva 3 tipkala. 50 je neka početna vrijednost, otprilike uzeta eksperimentalno.
```

```

void setup() { //Funkcija setup() poziva se kod započinjanja programa. Svaki program sastoji se od
setup i loop funkcija.
  pinMode(3, OUTPUT); // Pin D3 je izlaz (LED).
  pinMode(5, OUTPUT); // Pin D5 je izlaz (LED).
  pinMode(buttonPin, INPUT); //Pin D2 je ulaz (releji tipkalo žarulje).
  pinMode(releji, OUTPUT); //Pin D12 je izlaz (releji).
  pinMode(smjerPin, INPUT); //Pin D11 je ulaz (smjer tipkalo).
  pinMode(zaustaviPin, INPUT); //Pin D6 je ulaz (zaustavi/pokreni tipkalo).
  pinMode(13, OUTPUT); // Pin D13 je izlaz (LED na Arduino).
  digitalWrite(releji, ledState); //Pin D12 dobiva vrijednost HIGH ili LOW, zavisno je li releji
uključen. Postavlja se početno stanje releja.
}

```

```

void loop() {
  // read the sensor value za brzinu motora
  int sensorReading = analogRead(A2); // Čita razinu napona sa potencijometra (0-5V). Napon
pretvara u int raspona 0 do 1023.
  // map it to a range from 0 to 150:
  int motorSpeed = map(sensorReading, 0, 1023, 0, 150); //10-bitno, 0-5V postaje raspon brzine
vrtnje motora 0-150(RPM maksimalno).
  // set the motor speed:
  if (motorSpeed > 0) {
    myStepper.setSpeed(motorSpeed); // U moj stepper postavlja se brzina motora >0-150 RPM (1-
150).
  }
  tipkalo();
  pot1 = analogRead(A0); //Potencijometri za LED su spojeni na analogne pinove A0 i A1. Napon se
pretvara u int 0-1023.
  pot2 = analogRead(A1);
  if (pot1 > 21) { // Ako je napon veći od 21 što je 0,1V ( (5V/1024)*21=0,1025V ) jer je inace
nestabilan izlaz.
    led1 = map (pot1, 0, 1023, 0, 255); // Vrijednost pot1 raspona 0-1023 pretvara se u raspon 0-255.
  }
  else {
    pot1 = 0; // Ako je pot1 (niže u kodu i pot2) manji od 21 (0,1V) postavi se u 0 da se ne mapira u
neku vrijednost.
    led1 = 0; // U izlaz pošalji vrijednost 0 (ugasi LED).
  }
  if (pot2 > 21) { // Isto vrijedi i za pot2.
    led2 = map (pot2, 0, 1023, 0, 255);
  }
  else {
    pot2 = 0;
    led2 = 0;
  }
}

```



```

    smjerMotora(); // Pozivanje funkcije koja određuje smjer vrtnje motora.
    zaustavi(); // Pozivanje funkcije za zaustavljanje i pokretanje motora.
    analogWrite(3, led1); // PWM izlaz za led1.
    analogWrite(5, led2); // D3 i D5 postaju PWM izlazi koji prave signal koji aktivira tranzistore
    preko kojih se pak pale LED.
    //PWM simulira napon 0 i 5V neprestano (kvadratni signal) tako da LED svijetle (brzo titraju), a
    trajanje titraja ovisi o dužini gornjeg brida.
}

void tipkalo() { // Deklaracija funkcije tipkalo (za relej).
    // read the state of the switch into a local variable:
    int reading = digitalRead(buttonPin); // Čita stanje tipke za kontrolu releja (buttonPin (D2)) u
    varijablu int reading.

    // check to see if you just pressed the button
    // (i.e. the input went from LOW to HIGH), and you've waited
    // long enough since the last press to ignore any noise:

    // If the switch changed, due to noise or pressing:
    if (reading != lastButtonState) { // Ako je reading različit od zadnjeg stanja tipkala resetira se
    debounce tajmer.
        // reset the debouncing timer
        lastDebounceTime = millis(); // millis broji koliko ms radi ovaj program i trenutni broj upisuje u
        varijablu lastDebounceTime.
    }

    if ((millis() - lastDebounceTime) > debounceDelay) { // Očitava se novi broj millisa-a (trenutni
    broj milisekundi) i oduzima onaj broj upisan u lastDebounceTime. Ako je ta razlika veća od
    debounceDelay znači da je prošlo više od vremena potrebnog za debounce i trenutno stanje tipkala
    se može uzeti za stvarno stanje
        // whatever the reading is at, it's been there for longer
        // than the debounce delay, so take it as the actual current state:

        // if the button state has changed:
        if (reading != buttonState) { // Ako je trenutno očitavanje različito od zadnjeg upisanog trenutnog
        stanja
            buttonState = reading; // kao trenutno stanje upisuje se ono što je trenutno i očitano.

            // only toggle the LED if the new button state is HIGH
            if (buttonState == HIGH) {
                ledState = !ledState; // Promjeni stanje izlaza samo ako je stanje tipkala HIGH (ako je tipkalo
                stisnuto).
            }
        }
    }

    // Void tipkalo dio koda iznad uvijek je takav za tipkala. Služi za debounce tipkala. U ovom slučaju
    tipkala od releja.

```

```

// set the LED:
digitalWrite(relej, ledState); // Pali ili gasi žarulju preko releja.

// save the reading. Next time through the loop,
// it'll be the lastButtonState:
lastButtonState = reading; // Trenutno očitavanje upisuje se u lastButtonState. Kad se sljedeći put
prode kroz petlju ovo trenutno stanje bit će prošlo stanje.
}

int smjerMotora() { // Deklaracija smjerMotora tipkala D11. Dalje je kod isti samo se odnosi na
tipkalo smjera motora.
// read the state of the switch into a local variable:
int reading = digitalRead(smjerPin); // D11

// check to see if you just pressed the button
// (i.e. the input went from LOW to HIGH), and you've waited
// long enough since the last press to ignore any noise:

// If the switch changed, due to noise or pressing:
if (reading != lastSmjerButtonState) {
    // reset the debouncing timer
    lastDebounceTime1 = millis();
}

if ((millis() - lastDebounceTime1) > debounceDelay1) {
    // whatever the reading is at, it's been there for longer
    // than the debounce delay, so take it as the actual current state:

    // if the button state has changed:
    if (reading != smjerButtonState) {
        smjerButtonState = reading;

        // only toggle the LED if the new button state is HIGH
        if (smjerButtonState == HIGH) {
            smjer = !smjer; // Promjeni stanje izlaza samo ako je stanje tipkala HIGH (ako je tipkalo
stisnuto).
            stanje = LOW; // Kada je stanje LOW motor se okreće.
        }
    }
}

if (stanje == HIGH) { // Stanje je postavljeno na HIGH što predstavlja stanje mirovanja motora.
    myStepper.step(0); // a to znači da motor pravi 0 koraka.
    else if (smjer == HIGH) { // Ako je smjer motora HIGH to je smjer vrtnje kazaljke na satu.
        myStepper.step(1); // 1 je jedan korak u pozitivnom (clockwise) smjeru motora.
        digitalWrite(13, HIGH); // Pali se LED 13 na Arduino.
    }
}

```

```

else
{
  myStepper.step(-1); // -1 je jedan korak u negativnom smjeru motora.
  digitalWrite(13, LOW); // Gasi se LED 13 na Arduino.
}

// save the reading. Next time through the loop,
// it'll be the lastButtonState:
lastSmjerButtonState = reading; // Trenutno očitavanje upisuje se u lastSmjerButtonState.
}

void zaustavi() { // Deklaracija zaustavi tipkala pina D6. Dalje je sve isto
  // read the state of the switch into a local variable:
  int reading = digitalRead(zaustaviPin); // D6 (tipkalo).

  // check to see if you just pressed the button
  // (i.e. the input went from LOW to HIGH), and you've waited
  // long enough since the last press to ignore any noise:

  // If the switch changed, due to noise or pressing:
  if (reading != lastZaustaviState) {
    // reset the debouncing timer
    lastDebounceTime2 = millis();
  }

  if ((millis() - lastDebounceTime2) > debounceDelay2) {
    // whatever the reading is at, it's been there for longer
    // than the debounce delay, so take it as the actual current state:

    // if the button state has changed:
    if (reading != zaustaviButtonState) {
      zaustaviButtonState = reading;

      // only toggle the LED if the new button state is HIGH
      if (zaustaviButtonState == HIGH) {
        stanje = !stanje;
      }
    }
  }
  if (stanje == HIGH) { // Ako je stanje HIGH
    myStepper.step(0); // motor pravi 0 koraka.
  }
  // save the reading. Next time through the loop,
  // it'll be the lastButtonState:
  lastZaustaviState = reading;
}

```